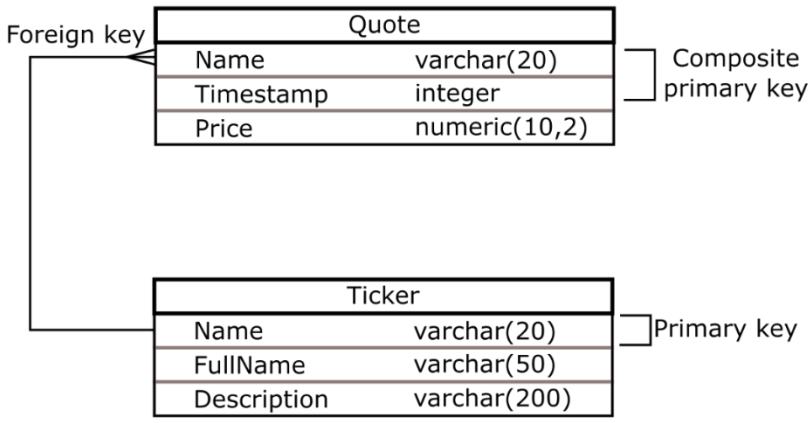

My PostgreSQL database consists of two tables. The table **Ticker** holds information about the stock tickers – its symbol(**Name**), full name(**Full name**) and a short description about this company. Exemplary row in the table can look like that: (ALE, Allegro, The biggest e-commerce platform in Poland.) The column **Name** is a primary key. **Ticker** table fulfills:

- First Normal Form - every cell in the table holds only single value and values in the single column are of the same type
- Second Normal Form – there is no partial dependency, because primary key consist of only one column
- Third Normal Form – there is no transitive dependency, because **FullName** and **Description** depends directly on **Name**.

The table **Quote** holds information about the stock quotes – ticker symbol(**Name**), particular timestamp (**Timestamp**) and the price for the given time(**Price**). We can insert plenty of stock quotes for the same ticker, that's why I used composite primary key consisting of **Name** and **Timestamp**. Column **Name** is the foreign key, because **Quote.Name** references to the **Ticker.Name** of the ticker. Exemplary row in the table can look like that: (ALE, 1624545244, 62.78). **Quote** table fulfills:

- First Normal Form - every cell in the table holds only single value and values in the single column are of the same type
- Second Normal Form – there is no partial dependency. Given price depends on both columns - the ticker symbol(**Name**) and time(**Timestamp**).
- Third Normal Form – there is no transitive dependency, **Price** depends directly on **Name** and **Timestamp**.

PostgreSQL indexes

In PostgreSQL, if we create a primary key the index is also created. So, by creating composite primary key(Name, Timestamp) the same index – index(Name, Timestamp) - is created. Thanks to that index, 'SELECT' queries looking for single quote or quotes with the same name will be fast even if there is a lot of data inside **Quote** table. If we want to use a lot of 'SELECT' queries looking for specific timestamp, index(Timestamp, Name) should be introduced to speed up scanning huge number of data.